IN THE CLAIMS:

Please amend the claims as follows. This listing of claims will replace all prior versions, and listings, of claims in the present application:

1. (Currently amended) A system for streaming a software application to a client comprising:

an application library having stored therein a prediction model and application files of the software application, the application files including executable code;

a streaming manager configured to send the application files to a client as a plurality of streamlets including executable code, each streamlet corresponding to a particular data block in a respective application file; and

a streaming prediction engine configured to identify at least one streamlet ~~which is predicted to be~~ as most appropriate to send to [[a]] the client at a particular time, based on an order in which code is predicted to be used when the software application is executed on the client, in accordance with the prediction model.

2. (Original) The system of claim 1, wherein each streamlet corresponds to a file data block having a size equal to a code page size used during file reads by an operating system expected to be present on a client system.

3. (Original) The system of claim 2, wherein the data block size is four kilobytes.

4. (Original) The system of claim 1, wherein the application files are stored in the application library as preprocessed streamlets, each streamlet corresponding to a data block in a particular application file at a particular offset and having a predefined length.

5. (Original) The system of claim 4, wherein the predefined length comprises a code page size used during file reads by an operating system expected to be present on a client system.

6. (Original) The system of claim 4, wherein each preprocessed streamlet is compressed.

7. (Original) The system of claim 1, wherein the streaming manager is configured to send the client upon a first initiation of the streaming application, a file structure specification of the application files.

8. (Original) The system of claim 7, wherein the streaming manager is further configured to send the client upon the first initiation of the streaming application a set of streamlets comprising at least those streamlets containing the portions of the application required to enable execution of the application to be initiated.

9. (Original) The system of claim 8, wherein the application library has a startup block comprising the file structure specification and set of streamlets stored therein.

10. (Original) The system of claim 1, wherein the streaming manager is further configured to install streaming environment support software on the client prior to initiating an application streaming processes.

11. (Original) The system of claim 1, further comprising a differential prediction model associated with the client, the prediction engine configured to make streamlet

predictions for the client in accordance with the default prediction model and the respective differential prediction model.

12. (Original) The system of claim 11, wherein the streaming manager is configured to, upon receipt of application usage tracking information from the client, update at least one of the differential prediction model for the client and the prediction model.

13. (Original) The system of claim 1, further comprising an application status repository comprising a data map for each active client, the data map generally indicating the streamlets which are present at the respective client.

14. (Original) The system of claim 13, wherein the streaming manager is configured to update the data map for the client upon a successful transmission of a streamlet to the client.

15. (Original) The system of claim 14, wherein the streaming manager is configured to, upon receipt of a request for a particular streamlet from the client:

determine if the data map indicates that the client already has the requested streamlet;

if the data map indicates that the requested streamlet is on the client system, request an updated data map from the client and replace the data map with a returned updated map;

retrieve the requested streamlet from the application library; and

update the data map upon a successful transmission of the requested streamlet to the client.

16. (Original) The system of claim 15, wherein the streaming manager is further configured to, upon receipt of the streamlet request from the client, reposition the prediction engine in the default prediction model in accordance with the requested streamlet.

17. (Original) The system of claim 13, wherein the streaming manager is configured to, upon receipt of an unsolicited data map from the client, replace the data map in the application status repository for the client with the data map received from the client.

18. (Original) The system of claim 17, wherein the streaming manager is further configured to, upon receipt of the unsolicited data map, compare the data map in the application status repository for the client with the data map received from the client and log mismatches.

19. (Currently amended) A method for streaming a software application comprising the steps of:

providing at a server an application library having stored therein application files of the software application, the application files including executable code;

forwarding the application files to a client as a particular sequence of streamlets including executable code, each streamlet corresponding to a particular data block in a respective application file; and

determining the particular sequence of streamlets in accordance with a prediction model indicating which streamlets are most appropriate to send to [[a]] the client at a particular time, based on an order in which code is predicted to be used when the software application is executed on the client.

20. (Original) The method of claim 19, wherein each streamlet corresponds to a file data block having a size equal to a code page size used during file reads by an operating system expected to be present on a client system.

21. (Original) The method of claim 20, wherein the data block size is four kilobytes.

22. (Original) The method of claim 19, further comprising the step of dividing the application files into streamlets prior to initiation of a streaming session.

23. (Original) The method of claim 19, further comprising the step of storing the application files in the application library as preprocessed streamlets, each streamlet corresponding to a data block in a particular application file at a particular offset and having a predefined length.

24. (Original) The method of claim 23, wherein the predefined length comprises a code page size used during file reads by an operating system expected to be present on a client system.

25. (Original) The method of claim 23, further comprising the step of compressing each streamlet prior to storage in the application library.

26. (Original) The method of claim 19, further comprising the step of sending the client upon a first initiation of the streaming application a file structure specification of the application files.

27. (Original) The method of claim 26, further comprising the step of sending to the client upon the first initiation of the streaming application a set of streamlets comprising at least those streamlets containing the portions of the application required to enable execution of the application to be initiated.

28. (Original) The method of claim 27, further comprising the step of storing in the application library a startup block comprising the file structure specification and set of streamlets stored therein.

29. (Original) The method of claim 19, further comprising the step of initiating a process to install streaming environment support software on the client prior to initiating an application streaming processes.

30. (Original) The method of claim 19, wherein the step of determining comprising determining the particular sequence of streamlets in accordance with the prediction model and a differential prediction model associated with the client.

31. (Original) The method of claim 30, further comprising the step of, upon receipt of application usage tracking information from the client, updating at least one of the differential prediction model for the client and the prediction model.

32. (Original) The method of claim 19, further comprising the steps of, upon receipt of a request for a particular streamlet from the client:

retrieving the requested streamlet from the application library; and

transmitting the streamlet to the client.

33. (Original) The method of claim 19, further comprising the steps of:

providing a data map for each active client generally indicating the streamlets which are present at the respective client; and

updating the data map associated with a particular client upon a successful transmission of a streamlet to the particular client.

34. (Original) The method of claim 33, further comprising the steps of, upon receipt of a request for a particular streamlet from the client:

determining if the data map associated with the client indicates that the already has the requested streamlet; and

in response to a positive determination, requesting an updated data map from the client and replacing the data map with a returned updated map.

35. (Original) The method of claim 34, further comprising the step of adjusting a position in the prediction model for the client in accordance with the requested streamlet.

36. (Original) The method of claim 33, further comprising the step of, upon receipt of an unsolicited data map from the client, replacing the data map in the application status repository for the client with the data map received from the client.

37. (Original) The method of claim 36, further comprising the steps of:

comparing the data map in the application status repository for the client with the unsolicited data map received from the client; and

logging mismatches identified during the comparing step.

38. (Currently amended) A computer program product stored on a computer readable medium, the product comprising a computer program for configuring a server with an application library having application files including executable code stored therein to stream a software application to a client, the computer program comprising code to configure the server to:

forward the application files to the client as a particular sequence of streamlets including executable code, each streamlet corresponding to a particular data block in a respective application file; and

determine the particular sequence of streamlets in accordance with a prediction model indicating which streamlets are most appropriate to send to the client at a particular time, based on an order in which code is predicted to be used when the software application is executed on the client.

39. (Original) The computer program product of claim 38, the computer program further comprising code to further configure the server to divide the application files into streamlets prior to initiation of a streaming session.

40. (Original) The computer program product of claim 39, the computer program further comprising code to configure the server to divide the application files into streamlets corresponding to a data block in a particular application file at a particular offset and having a predefined length.

41. (Original) The computer program product of claim 38, the computer program further comprising code to configure the server to send the client upon a first initiation of the streaming application a file structure specification of the application files.

42. (Original) The computer program product of claim 41, the computer program further comprising code to send to the client upon the first initiation of the streaming application a set of streamlets comprising at least those streamlets containing the portions of the application required to enable execution of the application to be initiated.

43. (Original) The computer program product of claim 42, the computer program further comprising code to store in the application library a startup block comprising the file structure specification and set of streamlets stored therein.

44. (Original) The computer program product of claim 38, the computer program further comprising code to install streaming environment support software on the client prior to initiating an application streaming processes.

45. (Original) The computer program product of claim 38, the computer program further comprising code to determine the particular sequence of streamlets in accordance with the prediction model and a differential prediction model associated with the client.

46. (Original) The computer program product of claim 45, the computer program further comprising code to, upon receipt at the server of application usage tracking information from the client, update at least one of the differential prediction model for the client and the prediction model.

47. (Original) The computer program product of claim 38, the computer program further comprising code to, upon receipt at the server of a request for a particular streamlet from the client:

retrieve the requested streamlet from the application library; and

transmit the streamlet to the client.

48. (Original) The computer program product of claim 38, the computer program further comprising code to:

provide a data map for each active client generally indicating the streamlets which are present at the respective client; and

update the data map associated with a particular client upon a successful transmission of a streamlet to the particular client.

49. (Original) The computer program product of claim 48, the computer program further comprising code to, upon receipt at the server of a request for a particular streamlet from the client:

determine if the data map associated with the client indicates that the already has the requested streamlet; and

in response to a positive determination, request an updated data map from the client and replacing the data map with a returned updated map.

50. (Original) The computer program product of claim 49, the computer program further comprising code to adjust a position in the prediction model for the client in accordance with the requested streamlet.

51. (Original) The computer program product of claim 48, the computer program further comprising code to, upon receipt at the server of an unsolicited data map from the client, replace the data map in the application status repository for the client with the data map received from the client.

52. (Original) The computer program product of claim 51, the computer program further comprising code to:

compare the data map in the application status repository for the client with the unsolicited data map received from the client; and

log mismatches identified during the comparing step.

53-55. (Canceled)

56. (New) A method comprising:

dividing at least a portion of a software application into blocks;

packaging the blocks into a repository from which the blocks can be individually extracted;

generating a predictive model for use in determining blocks likely to be loaded by the software application at various points during execution of the software application; and

generating an application package that includes the repository and the predictive model, the application package for use by a streaming server to stream the software application to a client.

57. (New) The method of claim 56, further comprising compressing each of the blocks prior to said packaging.

58. (New) The method of claim 56 wherein each of the blocks has a size equal to a code page size used during file reads by an operating system expected to be present on a system executing the software application.

59. (New) The method of claim 56, wherein the block size is four kilobytes.

60. (New) The method of claim 56, further comprising:

generating an index to the repository; and

including the index in the application package.

61. (New) The method of claim 56, wherein the software application has a file structure, the method further comprising including a specification of the file structure of the software application in the application package.

62. (New) The method of claim 56, wherein the repository is configured to permit access to blocks with reference to a source application file and an offset in the source application file.

63. (New) The method of claim 56, further comprising:

determining environmental changes introduced to a computer system by installation of the software application on the computer system; and

providing an environmental install package comprising data indicating the determined environmental changes for distribution.

64. (New) The method of claim 63, wherein said providing the environmental install package comprises including the environmental install package in the application package.

65. (New) The method of claim 56, further comprising:

determining a startup set comprising at least a minimal set of blocks sufficient for execution of the software application to be initiated; and

providing a startup streamlet set for distribution.

66. (New) The method of claim 65, wherein said providing the startup set comprises including the startup streamlet set in the application package.

67. (New) The method of claim 65, wherein said determining the startup set comprises:

executing the software application;

monitoring file block load requests at least until a designated startup point has been reached by the software application; and

identifying file blocks which were loaded prior to reaching the startup point.

68. (New) The method of claim 67, wherein the startup point comprises a point at which the software application waits for user input.